CLAIMS

What is claimed is:

SUBAR]

- 1. A method for gathering data from memory of a computer system, comprising the steps of:
- following a plurality of memory element
 descriptors of a machine readable record list to
- 5 locate data in the memory of the computer system,
- 6 where each memory descriptor is descriptive of data to
- 7 be retrieved from memory of the computer system;
- 8 gathering data specified by the plurality of
- 9 memory element descriptors; and
- 10 formatting the data into a buffer.
 - 1 2. The method for gathering data from memory of a
- 2 computer system of Claim 1, wherein at least one
- 3 memory descriptor is descriptive of a list memory
- 4 type, including location information of a head of a
- 5 list and tag information for at least one data element
- 6 to be gathered from a rode of the list.
- 1 3. The method for gathering data from memory of a
- 2 computer system of Claim 1, wherein at least one
- 3 memory descriptor is descriptive of a scalar memory
- 4 type.
- 1 4. The method for gathering data from memory of a
- 2 computer system of Claim 3, wherein at least one
- 3 memory descriptor is descriptive of a list memory
- 4 type, including location information of a head of a
- 5 list and tag information for at least one data element
- 6 to be gathered from a node of the list.

- 1 5. The method for gathering data from memory of a
- 2 computer system of Claim 3, wherein at least one
- 3 memory descriptor is a list memory descriptor,
- 4 including location information of a head of a first
- 5 list, location information of a head of a second list
- 6 in nodes of the first list, and tag information for at
- 7 least one data element to be gathered from nodes of
- 8 the second list.
- 1 6. A method for parsing a linked list to extract
- 2 data therefrom, the linked list stored in memory of a
- 3 computer system, comprising the steps of:
- 4 constructing a record list, the record list
- 5 comprising at least a first list element descriptor
- 6 descriptive of data to be retrieved from a first
- 7 linked list;
- 8 following a list head locator of the list element
- 9 descriptor to a head of the first linked list;
- 10 following links of the head of the first linked
- 11 list to a first node of the linked list;
- interpreting at least one tag of the first list
- 13 element descriptor to locate data of the node; and
- 14 extracting data from the node.
 - 1 7. The method of parsing a linked list of Claim 6,
- 2 wherein:
- 3 the record list further comprises a second list
- 4 element descriptor descriptive of data to be retrieved
- 5 from a second linked list, and wherein a node of the
- 6 first linked list contains a head of the second linked
- 7 list; and

- 8 the method further comprises the steps of:
- 9 following a list head locator of the second list
- 10 element descriptor to a secon¢ list head of the node
- 11 of the first linked list;
- following links of the /second list head to a node
- 13 of the second list;
- interpreting at least/one tag of the second list
- 15 element descriptor to locate data of the node of the
- 16 second list; and
- extracting data from the node of the second list
 - 1 8. The method of parsing a linked list of Claim 7,
 - 2 further comprising the step of formatting the
 - 3 extracted data into a capture buffer.
 - 1 9. The method of parsing a linked list of Claim 7
 - 2 further comprising the steps of
 - 3 stopping execution of all threads executing on
 - 4 the computer system/except for a thread parsing the
 - 5 list; and
- 6 resuming execution of all threads stopped during
- 7 the step of stopping execution;
- 8 wherein the step of stopping execution is
- 9 performed prior to the step of following links of the
- 10 head of the first linked list, and the step of
- 11 resuming execution is performed after the step of
- 12 extracting data from the node of the second list.

```
10. The method of Claim 9, further comprising the
1
2
    steps of:
         following links of the node of the second list to
3
    a second node of the second list, and
         extracting data from the second node of the
5
    second list according to at least one tag of the
6
    second list element descriptor.
7
    11. A computer system comprising at least one
1
    processor, a memory system, /and computer readable code
2
    recorded within the memory system, the code comprising
    computer readable code for
4
         receiving a record list, the record list
5
    comprising at least a first list element descriptor
6
    descriptive of data to be retrieved from a first
7
8
    linked list;
         following a list head locator of the list element
9
    descriptor to a head of the first linked list;
10
         following links of the head of the first linked
11
    list to a first node of the linked list;
12
         interpreting at least one tag of the first list
13
    element descriptor to locate data of the node; and
14
         extracting data form the node.
15
        The computer system of Claim 11, wherein:
1
    12.
         the record list further comprises a second list
2
    element descriptor descriptive of data to be retrieved
3
    from a second linked list, and wherein a node of the
 4
5
    first linked list contains a head of the second linked
```

list; and

```
7
         the computer readable code further comprises
8
    computer readable code for:
         following a list head locator of the second list
9
10
    element descriptor to a second/list head of the node
    of the first linked list;
11
12
         following links of the second list head to a node
13
    of the second list;
         interpreting at least one tag of the second list
14
    element descriptor to locate data of the node of the
15
16
    second list; and
17
         extracting data from the node of the second list
    13. A symbolic debugger for accessing data of named
1
2
    executable modules of an operating system executing on
    a target machine, the operating system having version
3
    information, the symbolic debugger comprising computer
4
    readable code stored on computer readable media, the
5
    computer readable code comprising code for
6
7
         a collection driver for execution on the target
    machine;
8
         a user interface capable of coupling to the
9
10
    collection driver; and
11
         a symbol resolution system capable of coupling to
12
    the user interface;
         wherein the user interface comprises computer
13
14
    readable code for constructing an input record list
15
    containing records describing data to be captured, at
16
    least some records of the input record list containing
17
    information derived from symbols resolved by the
```

\\Cs - 68854/5 - #39990 v4

- 18 symbol resolution system, and transmitting the input 19 record list to the collection driver;
- 20 wherein the collection driver further comprises
- 21 code for interpreting the input record list and
- 22 collecting operating system data into a capture buffer
- 23 specified by the input record list, and transmitting
- 24 the capture buffer to the user interface.
- 1 14. The symbolic debugger of Claim 13, wherein the
- 2 collection driver is capable of interpreting a record
- 3 of the input record list that specifies information to
- 4 be gathered from multiple nodes of a linked list.
- 1 15. The symbolic debugger of Claim 14, wherein the
- 2 collection driver is capable of interpreting a record
- 3 of the input record list that specifies information to
- 4 be gathered from multiple nodes of a linked list
- 5 having a list head located in a node of a parent list,
- 6 a list head of the parent list being specified by a
- 7 record of the input record list.
- 1 16. The symbolic debugger of Claim 14, wherein the
- 2 collection driver is capable of interpreting a record
- 3 of the input record list that specifies scalar
- 4 information to be gathered from designated locations
- 5 of the memory system.
- 1 17. The symbolic debugger of Claim 14, wherein the
- 2 collection driver further comprises a communications
- 3 interface capable of receiving the record list over a
- 4 network connection and comprises computer readable
- 5 code for reading the version information from the
- 6 operating system executing on the target machine.